Introduction to the IEEE Std. 1451.3-2003 Smart Transducer Standard

Lee H. Eccles
Boeing Commercial Airplane Company
Test & Data Systems Technology
P. O. Box 3707, M/C 14-ME
Seattle, WA 98124-2207
E-Mail lee.h.eccles@boeing.com

Abstract

The IEEE Std. 1451.3-2003 Smart Transducer Standard is a standard for smart transducers on a multi-drop transducer bus. This standard defines methods for providing data communications, time synchronization and DC power over a single pair of wires. This will lead to the ability to purchase COTS smart transducers that will plug into a data acquisition system and work with no further development effort. The standard does not define operation of the transducer or the signal conditioning but does describe how to take the digital output from the signal conditioning and to communicate that information to a host computer or node on a network. The standard describes the functionality of a smart transducer and the basic command set required to operate the transducer. It also provides the ability to extend the command set with additional commands such as the commands necessary to set up the signal conditioning. However, the feature that makes this standard, and other members of the IEEE 1451 family, different from other smart transducer standards is the use of Transducer Electronic Data Sheets or TEDS. The TEDS are used to provide the bus controller the ability to identify the transducers that are connected to the bus and to determine their operating characteristics. This leads to the ability to create data acquisition systems that are very nearly self-configuring which, greatly reduces the labor involved in setting up a data acquisition system.

KEY WORDS

Networks, Data Acquisition, Transducer Electronic Data Sheet, Metadata

INTRODUCTION

IEEE Std. 1451.3-2003 is a standard for interfacing digital output transducers to a local bus. It defines three different types of transducers; Sensors, Event Sensors and Actuators. Sensors sense some physical phenomena and present a digital representation of the magnitude of that phenomenon to a system. An event sensor detects a change of state in some physical phenomena and reports that change of state to the system. Actuators take a digital representation of the magnitude of some physical phenomena and use that information to control something. That being said this standard does not describe the transducers themselves or their signal conditioning but it describes how to interface a transducer to a digital communications network. In addition to describing how to interface the transducer it also provides Transducer Electronic Data Sheets or TEDS that describe to operating parameters of the Transducer and its signal conditioning.

The system concept

Figure 1 is a block diagram of a system containing a number of IEEE Std. 1451.3 transducer modules known as Transducer Bus Interface Modules or TBIMs. The sensor may be included in the TBIM or it may be external. The Transducer Bus, that is simply labeled Bus in the figure, connects these TBIMs together and to a bus controller. The bus has a linear rather than a star topology. Included in the unit containing the bus controller is the network interface. This device is referred to in the standard as a Network Capable Application Processor or NCAP. In a normal system there will be many NCAPs each with its own bus on a network. The data recorder, telemetry system and any other devices are connected to the network to allow access to all of the sensors in the system.

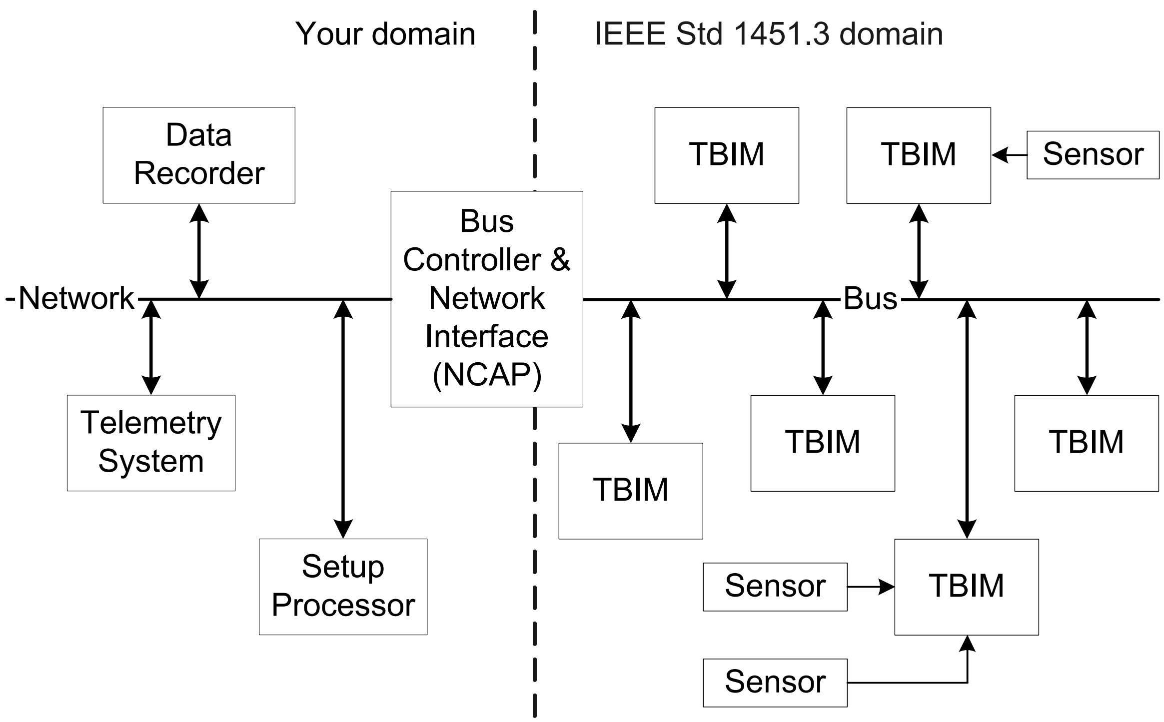

Figure 1 The System Concept

The Transducer Bus

The Transducer Bus is the physical media over which all communications between the bus controller and the transducer modules take place. There are three types of signals available on the bus. The most obvious type is the data. The second signal is a synchronization clock running at two MHz. The final one is dc power. All of these signals share a common twisted shielded pair of wires. This is accomplished using frequency division multiplexing. The data runs in a band between 4.75 and 9.25

megahertz. The synchronization clock runs at two megahertz and the power is at or near DC.

The following description discusses the operation of sensors since that is the transducer type most commonly used in T&E. However, it should be noted that the standard allows for actuators in which case the data will be flowing from the bus controller to the TBIM instead of from the TBIM to the bus controller. There are two ways in which the bus can be operated, command/response or streaming. For command/response operation the bus controller issues a trigger to one or more TBIMs on the bus that causes them to acquire one or more samples of data. After enough time has elapsed for the data to be acquired the bus controller issues a read data command to each TBIM that was triggered and reads back the data. This works well and is simple to understand but is not very bandwidth efficient. In the streaming mode of operation, time is divided into epochs. As shown in Figure 2, each epoch contains an isochronous interval and an asynchronous interval. The isochronous interval is further subdivided into 200 μsecond time slots. When the bus controller sets up each sensor, it assigns it a number of contiguous time slots in which to transmit its data. Then the bus controller issues a single trigger to the TBIM causing it to acquire a set of data samples and transmit them to the bus controller during its time slot. After that initial trigger the TBIM will acquire and transmit a set of data samples in each epoch without intervention of the bus controller. This streaming mode of operation requires the use of the two MHz synchronization clock to keep all TBIMs running synchronously. The asynchronous interval may be used to acquire data in the command/response mode or to manage the bus.

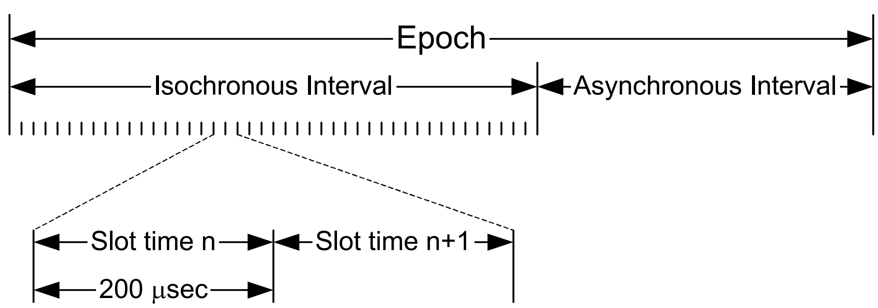

Figure 2 Bus Timing During Streaming Operation

An IEEE Std. 1451.3 sensor is intended to be used in a packet based data acquisition system rather than a conventional PCM system. In conventional IRIG 106 chapter 4 PCM systems a single sample of data is acquired and transmitted to the recorder and telemetry transmitter immediately. One of the big impacts that this has when using computer type communications networks is that the transmission overhead on the transducer bus consumes all of your bandwidth because are you trying to transmit a lot of one-word packets. This makes it desirable to collect several samples of data from each sensor and to transmit them in a single packet to preserve bandwidth. Yes, this does mean that there is some latency in the data. We have not found any way to avoid it in a packet-based system. However, the latency can be managed and kept within bounds. For low sample rate data it will often be necessary to transmit one data word packets but for higher sample rates the packets will be larger. The epoch defined above determines the maximum latency when operating in the streaming mode. The standard allows the epoch to be up to 250 milliseconds long. In the command/response mode of operation, the number of individual data samples acquired on each trigger and the sample rate determines the latency. The bus controller can increase the latency if it waits too long to read the data.

The design of the Transducer Bus is such that in either mode of operation it is possible to accurately time stamp the data samples in the bus controller or the TBIM. This does not eliminate the latency but it allows the data processing system to time align the data accurately for processing purposes. The time stamp must be applied in the bus controller in order to minimize the problems associated with the variable latency associated with transmitting the data over a commercial computer type network.

DATA FORMAT

IEEE 1451 assumes that all data is in eight bit bytes or octets when transmitted over the bus. With that as a limitation, a single data sample can require from one to 2,040 bits. There are seven different data models that can be used as listed in Table 1. There are N-octet Integers that may require up to eight octets and Long Integers that may contain up to 255. Long Integers will not be supported in most systems. N-Octet Fractions and Long Fractions are similar to the integer formats. With the integer formats, the least significant bit of the data is in the least significant bit of the word and any unused bits, which occupy the most significant bit positions, are set to zero. In the fractional representations the radix point is assumed to be between the two most significant bits of the word and any unused bits, which must be set to zero, are in the least significant bits of the word. The data can also be in a bit sequence for things like a bank of switches, single or double precision IEEE 754 floating point or in a Time-of-Day format. For a given transducer, the data model, the number of octets required and the number of significant bits in the word are given in the Transducer Electronic Data Sheet or TEDS. As we noted earlier with the Transducer Bus it is often desirable to collect more than one sample or each trigger or during a single epoch. The collections of samples are called data sets. A data set may contain from one to 65,535 samples of the data.

Table 1—Data Models

Data Model	Constraint on data model length
N-octet integer (unsigned)	$0 \le N \le 8$
Long integer (unsigned)	$9 \le N \le 255$
N-octet fraction (unsigned)	$0 \le N \le 8$
Long fraction (unsigned)	$9 \le N \le 255$
Single precision real	N=4
Double precision real	N=8
Bit sequence	1
Time of day	N=8

Another feature of the TBIM is that the manufacturer may choose to combine the output of several sensors into a single packet for transmission. This is accomplished by defining what is called a TransducerChannel proxy. A proxy has an address and can be read but it does not respond to many setup commands and does not have TEDS. There are two ways that the data from the sensors making up a proxy can be arranged as shown in Figure 3. If all data sets are the same length they can be interleaved. There is one variation on the interleaved method of combining data sets and that is that the interleaved data may be preceded by a single sample of another variable which would presumably be time. Otherwise they are arranged in blocks with the blocks of data from each sensor being appended together to form a larger block.

Setting up the measurement

The metadata required to set up the TBIM can come from a variety of sources and depends on just what features a manufacturer decides to build into the TBIM. For example, a TBIM can be built that contains a 10 g accelerometer. In this example, the TBIM becomes a self-contained unit. The sensor, signal conditioning and the A/D conversion are all built into the TBIM and are set up by the manufacturer. The user can simply plug it in and use it. This is considerably different from the usual process of setting up the signal conditioning to get the best resolution from the A/D converter and matching that to the sensor. To use this 10 g accelerometer in an 8 g application you would have the choice of deciding to accept the 20 % decrease in resolution or changing the set up of the TBIM. Assuming that the manufacturer has implemented the commands that are required to do the set up, you would send the unit into the Calibration lab and they would return an 8 g accelerometer. The sensor, signal conditioner and A/D converter would all be covered by the same calibration. Most of the metadata required for setup can be built into the TBIM and never seen by the user. However, there are two pieces of setup data that the user must supply. The first is the sample rate for the sensor. The sample rate can be fixed at the time of manufacture but that would be unusual in a vehicular instrumentation environment. There are provisions in the standard for a small non-volatile memory called the End User Application Specific TEDS that allows the user to store this type of information in the TBIM either in the lab or on the vehicle. The other piece of information that only the user can supply is the measurement title. The measurement title is not required to operate the transducer but it is convenient when using the installed TBIM. The Commissioning TEDS is a small non-volatile memory that is provided for this function. This same discussion would also apply if the transducer and the TBIM can both be sent into the Calibration Lab and calibrated as a pair.

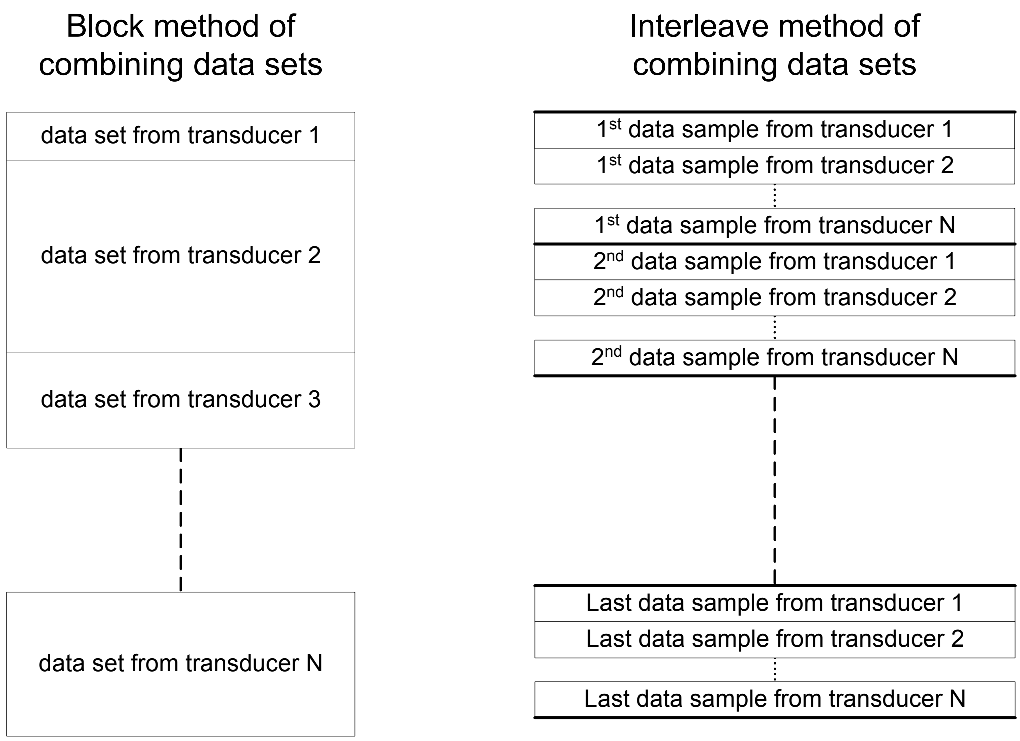

Figure 3 Combining Multiple Sensor outputs

There is also the other class of measurements where the Sensor and TBIM cannot be calibrated as a set. Structural strain gages and thermocouples are common examples of this type of measurements. For these types of measurements we need to consider a TBIM that is separate from the sensor and is calibrated separately from the sensor. Another way of looking at this is that the TBIM becomes a signal conditioner similar to the ones in use in present day systems. There are two different scenarios that we will consider in this paper. The first scenario is the traditional approach. We take the general-purpose signal conditioning that we have used in the past, integrate an A/D converter, if one is not already present, and add the bus interface. It works just like the device that we have used in the past and is just as expensive as it ever was. The approach at the other end of the cost spectrum is to use hardware that is not as costly but requires that it be set up for the measurement and calibrated for that specific application. An untrimmed analog ASIC will normally provide tolerances of around 20%. However, with a little bit of overkill in the A/D converter, i.e. higher resolution than absolutely required, this can be digitally compensated. The same mechanism that can digitally correct for component tolerances can also compensate for drifts caused by temperature. That mechanism is the Calibration TEDS and the correction method that it supports. In either case the setup for the TBIM is expected to be stored in non-volatile memory in the TBIM. Thus it can be done ahead of time in the laboratory and will retain the setup when installed on the vehicle. It can be changed on the vehicle if required.

TEDS

TEDS can be either embedded or virtual. Embedded TEDS are stored in non-volatile memory within the TBIM. Virtual TEDS are provided by the manufacturer but are stored someplace other than the TBIM. The response to a Query TEDS command will identify whether or not the TEDS are embedded or virtual. It is assumed that in most cases the manufacturer will furnish the virtual TEDS but that the user will store them in the users system.

There are twelve TEDS currently defined in the IEEE Std. 1451.3 standard. Of these, four are required to be implemented in any transducer module. The four required TEDS are:

- Module Meta-TEDS
- TransducerChannel TEDS
- Commissioning TEDS
- End User Application Specific TEDS.

General format for TEDS

All TEDS have the general format shown in Table 2. The first field in any TEDS is the TEDS length. It is a four octet unsigned integer. The length field is followed by a TEDS Identification Header. The next block is the information content for the TEDS. Depending upon the TEDS, the information may be binary information or it may be text-based. The last field in any TEDS is a checksum that is used to verify the integrity of the TEDS.

Table 2—Generic format for any TEDS

Field	Description	Type	# bytes
—	TEDS length	U32L	4
1 to N	Data block	Variable	Variable
—	Checksum	U16C	2

TEDS length

The TEDS length is the total number of bytes in the TEDS plus the two bytes in the checksum. This allows the number of bytes in the TEDS to be up to 4,294,967,296. Why a number that large? It is desired that the TEDS length field be the same for all TEDS and the text based TEDS have the potential to become large. A four byte integer was considered the smallest number that would provide enough length for any current usage and for several years into the future.

Data block structure

This structure contains the information that is stored in a specific TEDS. The fields that comprise this structure are different for each TEDS type.

Checksum

The checksum is the one's complement of the sum (modulo 2^{16}) of all preceding bytes, including the initial TEDS length field and the entire TEDS data block. The checksum calculation excludes the checksum field.

Meta-TEDS

This TEDS gives a few parameters that the system will need to know to be able to communicate reliably with the module, but its real purpose is to describe the relationships between the various transducers that are incorporated within that module. The first type of relationship is defined because it is expected that using actuators to control some aspects of the signal conditioning for a sensor will become common. For example an actuator might be used to set the sample rate for a sensor. In order for this to work, the system will need to know which actuator is associated with which sensor and what aspect of the sensor the actuator is controlling. The second type of relationship is defined because there are transducer modules that contain multiple transducers that have relationships that the user needs to know to be able to interpret the data. An example of this is a three axis accelerometer. Which transducer represents the x-axis, the y-axis and the z-axis. The third type of relationship between transducers is the Transducer Channel Proxy. Proxies are defined to provide a way for the module to combine the outputs from multiple sensors or the inputs to multiple actuators into a single entity to improve communications efficiency. The Meta-TEDS describes the proxy in terms of how the various transducer outputs are combined.

TransducerChannel TEDS

This TEDS defines the characteristics of a given TransducerChannel. It is divided into five major sections. The first section defines what is being measured and what range of operation the unit is designed to work over. It also tells whether or nor a Calibration TEDS is supplied and if the calibration can be applied within the module. The self test capability for the transducer is also identified. The second major section describes the sample of data that the transducer provides, if it is a sensor, or needs, if it is an actuator. This includes such information as whether the data is in an integer format or some other format as well as the number of samples that can be acquired with a single trigger. The third section contains the timing information for this transducer. With the information in this section the NCAP is able to compute the time of a sample, determine the time-out values needed to communicate with it, and determine the time required to run the self-test. The forth section lists the attributes associated with the transducer channel. The NCAP reads these attributes to determine the operating modes supported by the transducer. The last section is a few miscellaneous fields.

Commissioning TEDS

Every transducer in a system needs a user-friendly identification. This TEDS is intended to be used to give a transducer a name that is independent of the physical address of that transducer. For example, the transducer on NCAP number 4 in the transducer module with the alias of 17 and the channel number of 1 could be named "X-axis acceleration at BS 422." The content of a Commissioning TEDS is user defined. This TEDS is designed to be a text-based TEDS with space for 32 characters plus the header for a text based TEDS. If the user chooses to define it differently there are 42 bytes set aside for this TEDS.

End-User Application Specific TEDS

Like the Commissioning TEDS, the content of this TEDS is user defined. It is simply a place in non-volatile memory where the end user can place up to 256 bytes of information plus a length field and a checksum. The standard places no restrictions on what this may be used for. In one application that the author is aware of this is being used for storing the sample rate.

Optional TEDS

There are two groups of TEDS that the manufacturer may or may not choose to implement. The TEDS in the first group are intended to be used to calibrate or compensate the TransducerChannel. They are as follows:

Calibration TEDS – This TEDS contains the calibration constants necessary to convert the raw output of a sensor into engineering units or the input to an actuator from engineering units into the form needed by the actuator. The standard also defines the method that is to be used to process the data from its raw form into engineering units using the information from this TEDS.

Frequency Response TEDS – This TEDS gives the frequency response of the transducer along with its signal conditioning and any other processing done within the module in table form.

Transfer Function TEDS – This TEDS provides the same information as the Frequency Response TEDS except in an algorithmic form.

The second group of optional TEDS are Text-based. They are intended to provide information in a form that the operator can display and read. The TEDS described previously were intended for the use by the processors in the system but these TEDS are solely for the operator. The information in these TEDS is stored in XML format and may be in multiple languages. They are:

- Meta-Identification TEDS
- TransducerChannel Identification TEDS
- Calibration-Identification TEDS
- Location and Title TEDS
- Commands TEDS

The Commands TEDS is the mechanism that allows the manufacturer to define setup commands for the transducer or signal conditioner that are not part of the standard. To use these commands the user would need an XML enabled browser and a set of software that allowed picking fields from the display, forming the contents of the fields into the command format and transmitting the resulting command to the TBIM.

Commands

At the time of writing, IEEE p1451.0 defines ninety-four commands, many of which are optional depending upon what the manufacturer has implemented. The standard command format is divided into two eight bit fields. One of these fields is called the "Command Class" and the other the "Command Function." There are nine different command classes and each of them has functions associated with it.

The response to some of these commands is fully defined and others are only loosely defined. An example of a completely defined command would be Read TEDS Block. The loosely defined commands are things like sleep, reset or Calibrate transducer. For these commands, the general function to be performed is defined and the details are left to the manufacturer.

Initialization Command Class

A device only responds to commands in this class when in the halted state. The transducer will respond to one of these commands with an error message, if the system is set up for that to happen, but will otherwise ignore any of these commands that are issued when the device is not halted. There are sixteen commands in this class. As can be seen from Table 3 these commands tend to come in pairs. With the exception of the Calibrate and Zero commands, if the features one of these commands is intended to support is implemented then the other command is required.

Table 3—Initialization Commands

Sleep Wake-up	Set TransducerChannel data repetition count Read TransducerChannel data repetition count	Set TransducerChannel pre-trigger count Read TransducerChannel pre-trigger count
Calibrate TransducerChannel Zero TransducerChannel	Enable Corrections Disable Corrections	AddressGroup Definition Read AddressGroup assignment
Store operational setup Erase operational setup	Read TBIM structure	

Operational commands

Operational commands are the class of commands that are expected to be used in the collection and processing of data. These commands can be issued at any time, with the exception of when the transducer module is in the sleep state. Table 4 lists the commands in this class along with their rough groupings.

Table 4—Operational Commands

Query data block Read TransducerChannel data	Enable TransducerChannel trigger Disable TransducerChannel trigger
Read TransducerChannel data block Write TransducerChannel data Write TransducerChannel data block	Write service request mask Read service request mask Read Status
Read TIM version	Reset Halt

Set operating mode commands

When discussing the various transducer types we noted that there were different operating modes that could be associated with each of the transducer types. The capabilities of a given transducer to operate in one of these modes are defined in the TransducerChannel TEDS where they are called attributes. Table 5 lists the possible operating modes and the types of transducers that they apply to.

Table 5—Operating mode command functions

Function	Operational mode	Choices	Applies to
0	Reserved	—	
1	Sampling mode	Trigger-initiated	All
		Free-running without pre-trigger	
		Free-running with pre-trigger	
2	Buffered mode	Unbuffered	Sensors and actuators
		Buffered	
3	End-of-data-set operation	Hold	Actuators
		Recirculate	
4	Streaming mode	Off	All
		On	
5	Edge-to-report mode	Rising	Event Sensors
		Falling	
		both	
6	Actuator-halt	Hold	Actuators
		Finish data set	
		Ramp	
7 - 127	Reserved	—	
128 - 255	open to manufacturers	—	

Read operating mode commands

This command is used to read the operating mode as discussed in the previous paragraph.

Run Diagnostic commands

This is one of the very loosely defined commands. Per the standard it only has one argument and that is used to say to run all diagnostics. The manufacturer may add additional commands in this class and in most cases is expected to do so. In order to make them useful for the user this would require a Commands TEDS to identify them.

TEDS Access commands

There are four commands associated with the TEDS access, but that they can be used to access any of 255 different TEDS.

Query TEDS

The Query TEDS command is used by the NCAP to learn information about a specific TEDS. This command is a request for the transducer module to give information to the NCAP that is needed to transfer the TEDS between the transducer module and the NCAP.

Read TEDS block command

A read TEDS block command is used to read a block of information from the transducer module to the NCAP. The command is designed to allow the NCAP to be used to read the TEDS a block at a time depending upon how much data can be transferred at a time. It can also be used to read a portion if the TEDS.

Write TEDS block command

The Write TEDS block command is used to write the TEDS in a transducer module. It is very similar in function to the Read TEDS Block command.

At the first point in the processing of Write TEDS Block command that the data in non-volatile memory is altered, the TEDS is marked as invalid. It will remain marked as invalid, at least, up to the point that the Update TEDS command is received

Update TEDS command

The Update TEDS command is sent by the NCAP to indicate that the TEDS has been completely written. The transducer module will calculate the checksum for the entire TEDS and compare its value to the last two bytes of the TEDS. If the checksum is correct the TEDS may then be marked as Valid. If the checksum is wrong then the TEDS remains marked as invalid.

Manufacturer Defined commands

Some people believe that the working group went too far with the number of commands that have been defined but everybody realizes that what is defined is only a small subset of the total set of commands required to setup and control the various types of transducers and signal conditioners available on the market. To get around this problem but still allow a transducer manufacturer to build anything that is desired the standard allows the manufacturer to define additional commands. The commands should use the same structure as used in the commands defined in the standard. That is, they should have a command class and command function that is followed by any required arguments. In order to make these commands immediately available to the user, a Commands TEDS is used to describe these commands.

CONCLUSIONS

IEEE Std. 1451.3 is a new standard for smart sensors for use in a network based data acquisition system. This standard describes the characteristics of the sensor module and provides TEDS to provide detailed information about the module. However, it does not describe how the overall acquisition system should work. There are still many questions to be answered about the system. Some of those questions relate to just how we want to use the tools that the standard provides. We can stick with tried and true methods or we can investigate possible new architectures and processes. We may be able to automate features that we now do by hand and thus achieve considerable cost reductions. This has been a somewhat non-technical description of some of the possibilities. Hopefully it has given you some ideas and will open up new approaches to old problems for you.